

CLAIMS

I claim:

1. A method of reducing the number of times a tree data structure is rebalanced comprising the steps of:
 - (a) allowing a sub-tree of the data tree structure to grow unbalanced to a threshold level greater than one; and
 - (b) rebalancing the data tree structure when the threshold level is reached.
2. The method of claim 1 wherein the threshold level is $\log_2 n$ for a tree data structure having about n nodes.
3. The method of claim 1 wherein the threshold level is a constant number of levels greater than a level of a balanced portion of the tree data structure.

1 4. The method of claim 1 wherein the step of rebalancing the
2 tree data structure further comprises:

- 3 (a) developing first and second sets of rebalancing
4 operation tasks, the first set of operation tasks
5 operable to effect a first set of element state
6 transitions and the second set of operation tasks
7 operable to effect a second set of element state
8 transitions, the first and second set of element state
9 transition being distinct one from the other;
10 (b) performing the first set of operation tasks in a first
11 phase; and
12 (c) performing the second set of operation tasks in a
13 second phase.

14 5. A method of deferring the rebalancing of a tree data
15 structure comprising the steps of:

- 16 (a) allowing a sub-tree of the tree data structure to grow
17 unbalanced to a length greater than one; and
18 (b) rebalancing the tree data structure when the length of
19 the sub-tree reaches a threshold level.

20 6. The method of claim 5 wherein the threshold level is $\log_2 n$
21 for a tree data structure having about n nodes.

1 7. The method of claim 5 wherein the threshold level is a
2 constant number of levels greater than a level of a balanced
3 portion of the tree data structure.

1 8. A method of deferring the rebalancing of a tree data
2 structure comprising the steps of:

3 (a) tracking the performance of operations upon the tree
4 data structure; and

5 (b) rebalancing the tree data structure when an unbalanced
6 sub-tree of the tree data structure reaches a
7 threshold level greater than one.

8 9. The method of claim 8 wherein the threshold level is $\log_2 n$
9 for a tree data structure having about n nodes.

1 10. A method of performing a rebalancing operation upon a tree
2 data structure comprising the steps of:
3 (a) allowing a sub-tree of the tree data structure to grow
4 unbalanced to a threshold level greater than one;
5 (b) developing, in the case where the sub-tree reaches the
6 threshold level, first and second sets of rebalancing
7 operation tasks, the first and second set of
8 rebalancing operation tasks operable to effect a first
9 and second set of element state transitions
10 respectively;
11 (c) performing the first set of operation tasks in a first
12 phase; and
13 (d) performing the second set of operation tasks in a
14 second phase.

1 11. The method of claim 10 wherein the threshold level is $\log_2 n$
2 for a tree data structure having about n nodes.

1 12. A process for maintaining the balance of a tree data
2 structure comprising:
3 (a) monitoring a length of a sub-tree of the tree data
4 structure; and
5 (b) rebalancing the tree data structure when the length of
6 the sub-tree reaches a level greater than one.

1 13. A system for deferring the rebalancing of a tree data
2 structure comprising:
3 (a) a memory for storing the tree data structure; and
4 (b) a processor coupled to the memory, the processor
5 operable to track the performance of operations upon
6 the tree data structure and rebalance the tree data
7 structure when an unbalanced sub-tree of the tree data
8 structure reaches a threshold level greater than one.

1 14. The system of claim 13 wherein the threshold level is $\log_2 n$
2 for a tree data structure having about n nodes.

1 15. A system for deferring the rebalancing of a tree data
2 structure comprising:
3 (a) a means for storing the tree data structure; and
4 (b) a means for tracking the performance of operations
5 upon the tree data structure and rebalancing the tree
6 data structure when an unbalanced sub-tree of the tree
7 data structure reaches a threshold level greater than
8 one.

9 16. A computer readable medium for deferring the rebalancing of
10 a tree data structure, the computer readable medium comprising:

- 11 (a) a code segment for tracking the performance of
12 operations upon the tree data structure; and
13 (b) a code segment for rebalancing the tree data structure
14 when an unbalanced sub-tree of the tree data structure
15 reaches a threshold level greater than one.
16
17

1 17. A method of deferring the rebalancing of a tree data
2 structure comprising the steps of:
3 (a) tracking the performance of operations upon the tree
4 data structure; and
5 (b) rebalancing the tree data structure when an unbalanced
6 sub-tree of the tree data structure reaches a threshold level
7 greater than one, the rebalancing further comprising creating a
8 first set of rebalancing operation tasks, the first set of
9 rebalancing operation tasks being characterized by navigation of
10 the tree data structure using at least an existing link,
11 creating a second set of rebalancing operation tasks, the second
12 set of rebalancing operation tasks being different from the
13 first set of rebalancing operation tasks and being characterized
14 by location of elements within the tree data structure using at
15 least one pointer created by the first set of rebalancing
16 operation tasks, and performing at least one operation task of
17 the first set of rebalancing operation tasks in a first phase
18 and at least one of the second set of rebalancing operation
19 tasks in a second phase.

1 18. A method of deferring the rebalancing of a tree data
2 structure comprising the steps of:
3 (a) tracking the performance of operations upon the tree
4 data structure; and
5 (b) rebalancing the tree data structure when an unbalanced
6 sub-tree of the tree data structure reaches a threshold level
7 greater than one, the rebalancing further comprising executing
8 simultaneous rebalancing operations on the tree data structure
9 including performing any first phase operation task of each of
10 the simultaneous rebalancing operations in a first phase using
11 parallel processes, developing a set of serial rebalancing
12 operations during the first phase, and performing any second
13 phase operation task of each of the simultaneous rebalancing
14 operations in a second phase, the second phase operation task
15 having at least one of the set of serial rebalancing operations.